ECSE−4750 Computer Graphics HW2

Ying Lu

September 11, 2013

1. Which RPI grad was the technical person in the founding group of NVidia?

   Solution: Curtis Priem, who received a B.S. degree in electrical engineering from RPI in 1982, cofounded NVIDIA.

2. What hardware component had to get much cheaper in order to make frame buffers possible?

   Solution: Memory has to get much cheaper in order to make frame buffers possible.

3. What is the main physical principal that makes the Kinect function? Write not more than 100 words. This requires research. There may be several valid answers.

   Solution:

   (a) The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions

   (b) The RGB camera, depth sensor and multi-array microphone running proprietary software provide full-body 3D motion capture, facial recognition and voice recognition capabilities

   (c) The Kinect sensor's microphone array enables the Xbox 360 to conduct acoustic source localization and ambient noise suppression

4. Modify the simpleViewer demo program from last week to display the spaceship NCC1701 as follows.

(a) Strip out the code to draw the spiral.

(b) In the initialization routine, read a list of triangles from the file ncc1701b.data. Each line gives the vertices of one triangle as follows: x1 y1 z1 x2 y2 z2 x3 y3 z3

(c) Alternatively, if you have too much time and want to test the limits of your compiler, use your favorite powerful editor to turn that file into a several thousand line C++ initialization statement, embed it into your program, and then see if it will compile.)

(d) Draw each triangle with a color whose brightness depends on the angle between the triangle and the plane z=0. (This will approximate a light shining from up high.)

(e) Put all the vertices into one array and the colors into another and then draw everything with one call.

Solution:

(a) Done following the instructions online

(b) Done with the instructions online

(c) N/A

(d) Done

(e) For this part, in order to reflect that the color whose brightness depends on the angle between the triangle and the plane $z = 0$, I use two ways of assigning the color to the *glColor3f*, the first one is:

```
glColor3f(0.3, 0.5, 0.3 + angle);
```

The screen copy from the top is:

The screen copy from the bottom is:

The second one is:

```
glColor3f(0.3+angle, 0.3 + angle, 0.3 + angle);
```

The screen copy from the top is: The screen copy from the bottom is:

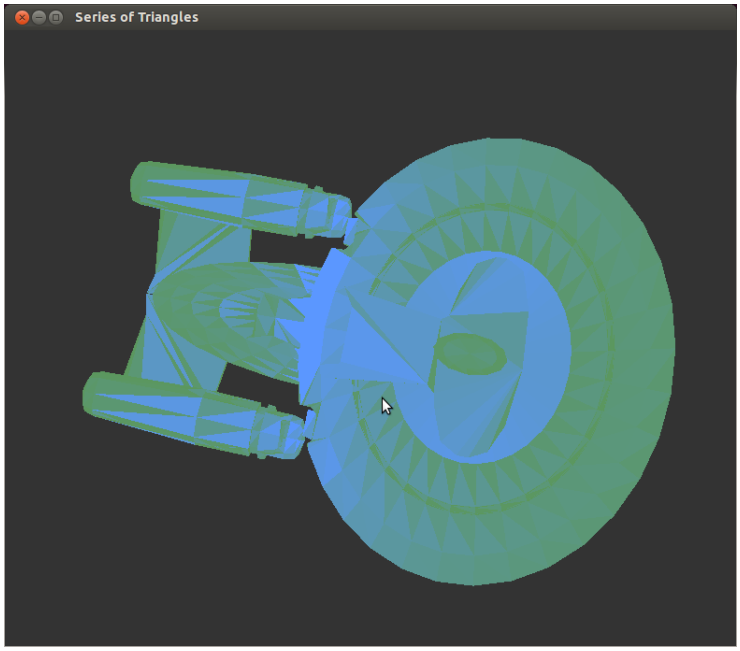

Figure 1: The screen copy of the triangles viewed from top

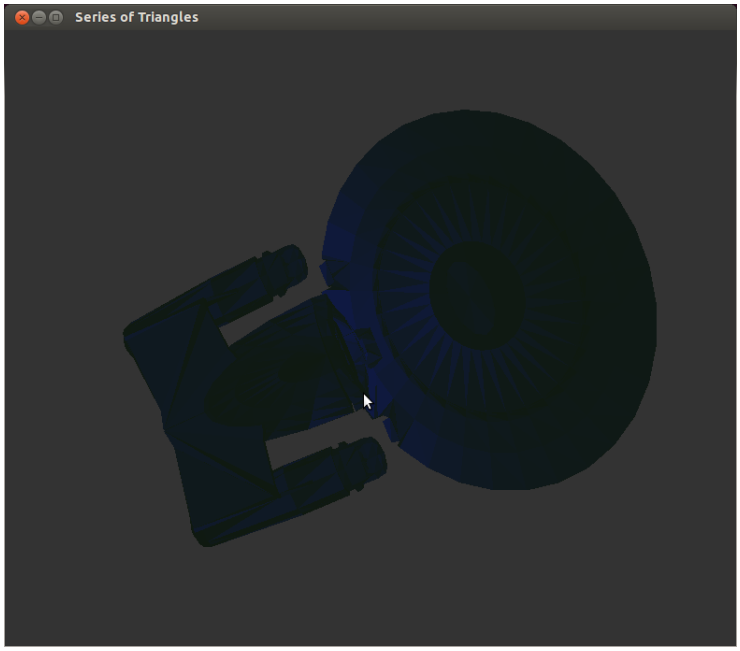

Figure 2: The screen copy of the triangles viewed from bottom

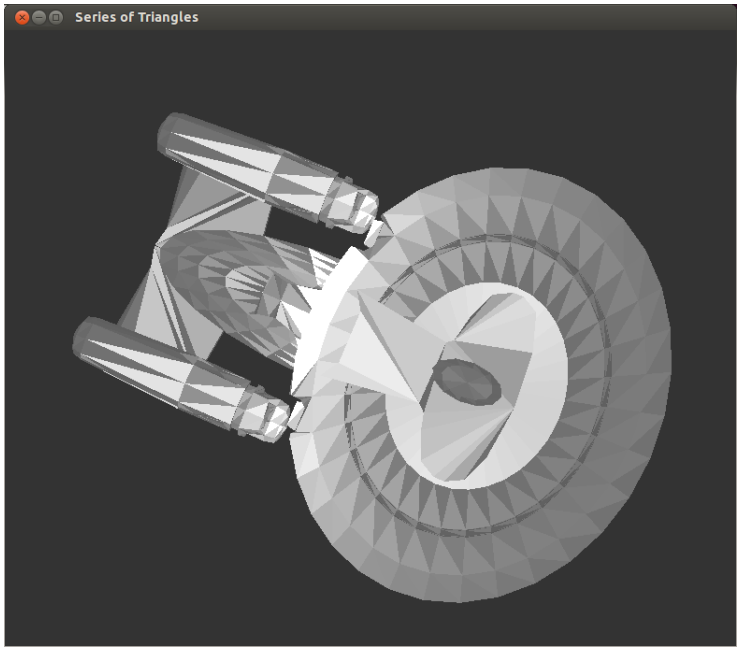

Figure 3: The screen copy of the triangles viewed from top

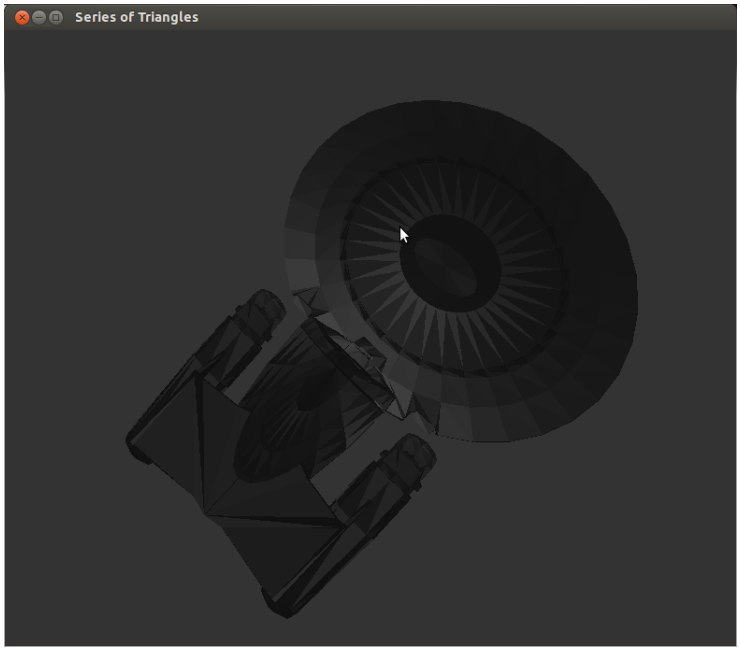

Figure 4: The screen copy of the triangles viewed from bottom